UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/042,581	01/09/2002	Thanh V. Lam	POU920010124US1	6963

23334 7590 09/22/2005

FLEIT, KAIN, GIBBONS, GUTMAN, BONGINI
& BIANCO P.L.
ONE BOCA COMMERCE CENTER
551 NORTHWEST 77TH STREET, SUITE 111
BOCA RATON, FL 33487

EXAMINER	
TAN, ALVIN H	

ART UNIT	PAPER NUMBER
2173	

DATE MAILED: 09/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/042,581	LAM ET AL.	
	Examiner	Art Unit	
	Alvin H. Tan	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 August 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3,4,6,7,9,11-13,15-17,19,20,22,23 and 25-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3,4,6,7,9,11-13,15-17,19,20,22,23 and 25-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 August 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Claims 1, 3-4, 6-7, 9, 11-13, 15-17, 19-20, 22-23, and 25-32 have been examined and rejected. This Office action is responsive to the amendment filed on 8/1/05, which has been entered in the above identified application.

Drawings

2. The drawings were received on 8/1/05. These drawings are acceptable.

Specification

3. The corrections to the specification have been approved, and the objections to the specification are withdrawn.

Claim Objections

4. The corrections to claims 9 and 13 have been approved, and the objections to those claims are withdrawn.

5. Claims 26 and 28 are objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. Claims 26 and 28

fails to further limit the subject matter because the independent claims already state the concurrent execution of at least one computer operating system command.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 26 and 28 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite in that it fails to point out what is included or excluded by the claim language. This claim is an omnibus type claim.

a. Claim 26 recites, "wherein the step of sending includes substantially simultaneously executing the selection". The term "substantially simultaneously" does not give a clear definition as to the time difference allowed between the execution of the command on the plurality of remote computers.

b. Claim 28 recites, "wherein the command dispatcher causes substantially simultaneous execution". The term "substantially simultaneously" does not give a clear definition as to the time difference allowed between the execution of the command on the plurality of remote computers.

Claim Rejections - 35 USC § 103

8. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

9. Claims 1, 3-4, 6-7, 9, 11-12, 15-17, 19-20, 22-23, 26-28, and 30-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maddocks et al (Pub. No. US2004/0201627 A1) and Bonnell et al (US Patent No 5,655,081).

Claims 1, 3-4, 6-7, 26 (Method)

Claims 9, 11-12, 15-16, 27-28 (System)

Claims 17, 19-20, 22-23 (Computer Readable Medium)

9-1. Regarding claims 1, 9, 27, and 17, Maddocks teaches the claimed invention by disclosing a "tree-based GUI that enables test programs for testing various devices" *[paragraph 15, lines 1-3]*. Maddocks teaches the claim for a method comprising pre-determining a command script file containing an operator defined plurality of computer operating system commands, by teaching, "All of the information needed to enable tests to be set up and executed may be stored in a memory device comprised by a computer" *[paragraph 17, lines 1-3]*. This would "enable multiple users to create, edit, and/or execute machine control sequences at their workstations by accessing previously created machine control sequences" *[paragraph 17, lines 11-14]*. Since all of the information needed to enable tests is stored, users are able to retrieve any part of the information needed to create their own tests. Hence, the commands associated with the devices when the device node is selected *[paragraph 22, lines 1-3]* are stored on a data

file and are pre-determined. In this way, the GUI pre-determines a command script file containing an operator defined plurality of computer operating system commands.

The command script file is displayed as shown *[figure 3, reference character "35"]* when the user selects a device node by clicking the left mouse button once *[paragraph 22, lines 1-3]*. The file contains a plurality of textual representations, each textual representation corresponding to one computer operating system command from among the plurality of computer operating system commands of the command script file.

Maddocks teaches the method comprising selecting a selection of at least one computer operating system command from among the plurality of computer operating system commands of the command script file, by teaching that the user can select commands and add them to the tree by highlighting the selected command from the Jlist "35" *[figure 3]* and then using the "ADD" button *[paragraph 22, lines 4-5]*.

Maddocks teaches executing the at least one computer operating system command contained within the selection, by teaching that the selected commands are executed when the user selects the "Run Sequence" tab and then selects the machine control sequence to be executed *[paragraph 28, lines 1-4]*.

While Maddocks teaches accessing machine control sequences over a network, Maddocks does not expressly teach sending the selection of at least one computer operating system command to a plurality of remote computers communicatively coupled to the local computer via a computer network for concurrent execution of the selection

of at least one computer operating system command on at least two of the plurality of remote computers.

Bonnell teaches, "Many business entities have one client/server network installed in each regional office, in which a high-capacity computer system operates as the "server" supporting many lower-capacity "client" desktop computers. The servers in such a business entity are also commonly connected to one another by a high-level network known as a wide area network. In this manner, users at any location within the business entity can theoretically access resources available in the company's network regardless of where the resource is located. The flexibility gained for users with this type of arrangement comes with a price, however. It is very difficult to manage such a diverse and widely-dispersed network for many reasons. Servers installed in the wide area network are frequently not all of the same variety. One regional office may be using an IBM machine with a UNIX operating system, while another regional office may be using a DEC machine with a VMS operating system" *[column 1, lines 27-44]*. Bonnell gives a solution to this problem by disclosing a method for managing a computer network *[figure 1]* where a network management computer system is coupled via a network to a server computer system and a plurality of other server computer systems. The hardware present in each of the computer systems need not be the same *[column 1, lines 54-61]*. Further, a command execution manager is responsible for coordinating the execution of commands. A run queue scheduler maintains a list of runable jobs or commands, together with the times at which they should be run and their desired frequency *[column 3, lines 19-29]*.

Since Maddocks' invention is portable over many operating systems and multiple users are able to access previously created tests, it would have been obvious to one of ordinary skill in the art at the time the invention was made to control the execution of commands, as taught by Maddocks, so that they would be executed on a plurality of remote computers as suggested by Bonnell, to allow for the execution of operating system commands to be performed on a plurality of remote computers. In this way, widely dispersed networks may be more easily managed by allowing the results of machine control sequence to be quickly and easily analyzed from a central console as suggested by Bonnell. Thus, businesses will get the benefit of allowing users at any location within the business entity to access resources available in the company's network regardless of where the resource is located while reducing the difficulty of having to manage a diverse network.

Bonnell and Maddocks do not expressly teach concurrent execution of commands on at least two of a plurality of remote computers. However, Examiner takes Official Notice that it is well known that performing the same process on two remote computers concurrently would be faster than performing the same process on one remote computer, and waiting for that process to be finished before performing the process on the next remote computer.

In the case of a network management computer managing commands to be executed on a plurality of server computer systems as taught by Maddocks and Bonnell, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have each of the plurality of remote computers execute the given

commands simultaneously rather than have each remote computer execute the commands separately, since Examiner takes Official Notice that performing a process concurrently is more efficient than performing a process one after the other.

9-2. Regarding claims 3, 11, and 19, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-1. Maddocks teaches the method wherein executing comprises single stepping through each computer operating system command within the selection, by further disclosing, "The Jtree is a hierarchical tree structure of sequences, steps, devices, and commands" *[paragraph 19, lines 9-10].* Maddocks also teaches "a particular machine control sequence is executed when the user selects the "Run Sequence" tab shown *[figure 3]* and then selects the machine control sequence to be executed. Thus, a user can single step through each command within the selection of commands if the user were to create multiple sequences, each with a single step, device, and command selected from the plurality of commands in Jlist "35" *[figure 3]*, and then execute each sequence.

9-3. Regarding claims 4, 12, and 20, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-1. Maddocks teaches the method of displaying a program output and an error output within separate displays, wherein the separate displays are at least one of separate GUI windows and separate display screens, by further disclosing, "Once the "Sequence Results" window "72" has been opened, an item "76" and value "77" *[figure 6]* pair are displayed. Underneath this pair, a number of

statistics are displayed that provide the user with a high-level summary of the execution results of the sequence for the entire sequence run" *[paragraph 30, lines 1-5]*. This summary of execution results represents the program output. "Whether the command succeeded or failed is indicated under labels "92" and "93" *[figure 6]* in the line in which the corresponding command is shown" *[paragraph 32, lines 4-6]*. This is where the error output is displayed.

Maddocks teaches "The "Sequence Results Viewer" portion of the GUI is a multiple document interface (MDI), which means that it allows multiple set of results, i.e., multiple files, to be simultaneously displayed in different "Sequence Results" windows *[paragraph 29, lines 1-5]*. The sequence results windows "72" and "75" *[figure 6]* may either be displayed as separate windows or as active portions of a different window so that they are simultaneously and fully viewable by the user *[paragraph 29, lines 22-25]*. Thus, the user may be able to open two separate GUI windows with the same sequence results and view the program output in one of them and the error output in the other.

9-4. Regarding claims 6, 15, and 22, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-1. Maddocks further teaches the method wherein executing comprises detecting a failure to execute the at least one computer operating system command, by teaching, "Whether the command succeeded or failed is indicated under labels "92" and "93" *[figure 6]* in the line in which the corresponding command is shown" *[paragraph 32, lines 4-6]*. Maddocks does not expressly teach the

plurality of remote computers on which the detecting of a failure to execute a command is done on. Bonnell does teach the advantage of running a monitoring system on a plurality of computers as stated in section 9-1. Since Maddocks' invention is portable over many operating systems and multiple users are able to access previously created tests, it would have been obvious to one of ordinary skill in the art at the time the invention was made to allow for the detection of a failure to execute at least one computer operating system command, as taught by Maddocks, so that they would be detected on the plurality of processors as suggested by Bonnell, to allow for the detection of a failure to execute at least one operating system command to be performed on a plurality of remote computers. In this way, widely dispersed networks may be more easily managed by allowing the results of machine control sequence to be quickly and easily analyzed from a central console as suggested by Bonnell. Thus, businesses will get the benefit of allowing users at any location within the business entity to access resources available in the company's network regardless of where the resource is located while reducing the difficulty of having to manage a diverse network.

In addition, Maddocks does not expressly teach a method wherein the executing comprises determining at least one remote computer within the plurality of remote computers that is unavailable for executing the at least one computer operating system command. Bonnell teaches a flow diagram *[figure 9]* illustrating a procedure for monitoring resources according to the network management system *[column 8, lines 13-15]*. Reference character "144" *[figure 9]* shows that a manager software sends a signal to the agent to initiate a monitoring procedure. If the agent is unavailable, the

procedure stops and the operating system commands will not execute *[reference character "152"]*. In this way, the manager software determines whether or not the agent is available for the execution of operating system commands. Bonnell also teaches that it is desirable in a large network to use numerous network management computer systems, each running its own manager software and to have agent processes in the network numbering in the thousands *[column 5, lines 37-41]*. Maddocks teaches that a need exists for a convenient and expedient way to view and analyze machine control sequences and the corresponding execution results *[paragraph 4, lines 1-3]*. By first sending a signal to the agent before execution begins, the system of Bonnell would be able to predetermine which agents are available for execution of commands and which are not. This would allow only the monitoring of agents that are available, which would result in a more organized and expedient method for analyzing machine control sequences over the network. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Maddocks with the teachings of Bonnell so that the analysis of machine control sequences could be run on a plurality of remote computers wherein the system determines which remote computers are available before execution.

9-5. Regarding claims 7, 16, and 23, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-4. Bonnell teaches that a knowledge database of the manager software system *[figure 2, reference character "47"]* may contain setup commands *[figure 4, reference character "84"]*. "Setup commands are those that are to

be executed whenever the manager software system establishes a connection with an agent software system" *[column 4, lines 21-23]*. Thus, when the manager software determines whether a processor is available or not, it is effectively requesting the execution of the commands. This claim shares the same motivation as was stated in section 9-4 above.

9-6. Regarding claims 26 and 28, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-1. The claim fails to further limit the subject matter of the previous claim and thus, shares the same motivation used to reject claim 1.

Claims 30-32

9-7. Regarding claim 30, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-1. Maddocks teaches the claimed invention by disclosing a "tree-based GUI that enables test programs for testing various devices" *[paragraph 15, lines 1-3]*. Maddocks teaches the claim for a method comprising pre-determining a command script file containing an operator defined plurality of computer operating system commands, by teaching, "All of the information needed to enable tests to be set up and executed may be stored in a memory device comprised by a computer" *[paragraph 17, lines 1-3]*. This would "enable multiple users to create, edit, and/or execute machine control sequences at their workstations by accessing previously created machine control sequences" *[paragraph 17, lines 11-14]*. Since all of the

information needed to enable tests is stored, users are able to retrieve any part of the information needed to create their own tests. Hence, the commands associated with the devices when the device node is selected *[paragraph 22, lines 1-3]* are stored on a data file and are pre-determined. In this way, the GUI pre-determines a command script file containing an operator defined plurality of computer operating system commands.

The command script file is displayed as shown *[figure 3, reference character "35"]* when the user selects a device node by clicking the left mouse button once *[paragraph 22, lines 1-3]*. The file contains a plurality of textual representations, each textual representation corresponding to one computer operating system command from among the plurality of computer operating system commands of the command script file.

Maddocks teaches the method comprising selecting a selection of at least one computer operating system command from among the plurality of computer operating system commands of the command script file, by teaching that the user can select commands and add them to the tree by highlighting the selected command from the Jlist "35" *[figure 3]* and then using the "ADD" button *[paragraph 22, lines 4-5]*.

Maddocks teaches executing the at least one computer operating system command contained within the selection, by teaching that the selected commands are executed when the user selects the "Run Sequence" tab and then selects the machine control sequence to be executed *[paragraph 28, lines 1-4]*.

While Maddocks teaches accessing machine control sequences over a network, Maddocks does not expressly teach sending the selection of at least one computer

operating system command to a plurality of remote computers communicatively coupled to the local computer via a computer network for concurrent execution of the selection of at least one computer operating system command on at least two of the plurality of remote computers.

Bonnell teaches, "Many business entities have one client/server network installed in each regional office, in which a high-capacity computer system operates as the "server" supporting many lower-capacity "client" desktop computers. The servers in such a business entity are also commonly connected to one another by a high-level network known as a wide area network. In this manner, users at any location within the business entity can theoretically access resources available in the company's network regardless of where the resource is located. The flexibility gained for users with this type of arrangement comes with a price, however. It is very difficult to manage such a diverse and widely-dispersed network for many reasons. Servers installed in the wide area network are frequently not all of the same variety. One regional office may be using an IBM machine with a UNIX operating system, while another regional office may be using a DEC machine with a VMS operating system" *[column 1, lines 27-44]*. Bonnell gives a solution to this problem by disclosing a method for managing a computer network *[figure 1]* where a network management computer system is coupled via a network to a server computer system and a plurality of other server computer systems. The hardware present in each of the computer systems need not be the same *[column 1, lines 54-61]*. A command execution manager is responsible for coordinating the execution of commands. An object database manager creates and maintains a

database representing all of the resources and applications present on the computer network, as well as information pertaining to the state of those objects, in a form that will be readily usable by a graphical user interface module. A graphical user interface is responsible for communicating with display driver software in order to present visual representations of objects on the display of network management computer system. Such representations typically take the form of icons for objects. Also, the graphical user interface module coordinates the representation of pop-up windows for command menus and the display of requested or monitored data *[column 2, lines 36-51; figure 2].* Further, a communications module on the management system is responsible for handling communications to and from agent software systems installed throughout the computer network *[column 2, line 67; column 3, lines 1-2].* Thus, the management system is presented with a list of all the remote computers on the network and is able to monitor and control communications between each of them. A script program compiler is responsible for compiling script programs which are interpretable by script program interpreters located on each of the remote computers *[column 3, lines 15-19].* A run queue scheduler maintains a list of runable jobs or commands, together with the times at which they should be run and their desired frequency *[column 3, lines 19-29].* Thus, once the remote computers receive the program script, the remote computers execute the commands independently of each other.

Since Maddocks' invention is portable over many operating systems and multiple users are able to access previously created tests, it would have been obvious to one of ordinary skill in the art at the time the invention was made to control the execution of

commands, as taught by Maddocks, so that they would be executed on a plurality of remote computers as suggested by Bonnell, to allow for the execution of operating system commands to be performed on a plurality of remote computers. In this way, widely dispersed networks may be more easily managed by allowing the results of machine control sequence to be quickly and easily analyzed from a central console as suggested by Bonnell. Thus, businesses will get the benefit of allowing users at any location within the business entity to access resources available in the company's network regardless of where the resource is located while reducing the difficulty of having to manage a diverse network.

Bonnell and Maddocks do not expressly teach concurrent execution of commands on at least two of a plurality of remote computers. However, Examiner takes Official Notice that it is well known that performing the same process on two remote computers, wherein the process is executed independently of each of the computers, concurrently would be faster than performing the same process on one remote computer, and waiting for that process to be finished before performing the process on the next remote computer.

In the case of a network management computer managing commands to be executed on a plurality of server computer systems as taught by Maddocks and Bonnell, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have each of the plurality of remote computers execute the given commands simultaneously rather than have each remote computer execute the

commands separately, since Examiner takes Official Notice that performing a process concurrently is more efficient than performing a process one after the other.

9-8. Regarding claim 31, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-7. Maddocks further teaches the method wherein executing comprises detecting a failure to execute the at least one computer operating system command, by teaching, "Whether the command succeeded or failed is indicated under labels "92" and "93" *[figure 6]* in the line in which the corresponding command is shown" *[paragraph 32, lines 4-6]*. Maddocks does not expressly teach the plurality of remote computers on which the detecting of a failure to execute a command is done on. Bonnell does teach the advantage of running a monitoring system on a plurality of computers as stated in section 9-7. Since Maddocks' invention is portable over many operating systems and multiple users are able to access previously created tests, it would have been obvious to one of ordinary skill in the art at the time the invention was made to allow for the detection of a failure to execute at least one computer operating system command, as taught by Maddocks, so that they would be detected on the plurality of processors as suggested by Bonnell, to allow for the detection of a failure to execute at least one operating system command to be performed on a plurality of remote computers. In this way, widely dispersed networks may be more easily managed by allowing the results of machine control sequence to be quickly and easily analyzed from a central console as suggested by Bonnell. Thus, businesses will get the benefit of allowing users at any location within the business entity to access resources available in

the company's network regardless of where the resource is located while reducing the difficulty of having to manage a diverse network.

In addition, Maddocks does not expressly teach a method wherein the executing comprises determining at least one remote computer within the plurality of remote computers that is unavailable for executing the at least one computer operating system command. Bonnell teaches a flow diagram *[figure 9]* illustrating a procedure for monitoring resources according to the network management system *[column 8, lines 13-15]*. Reference character "144" *[figure 9]* shows that a manager software sends a signal to the agent to initiate a monitoring procedure. If the agent is unavailable, the procedure stops and the operating system commands will not execute *[reference character "152"]*. In this way, the manager software determines whether or not the agent is available for the execution of operating system commands. Bonnell also teaches that it is desirable in a large network to use numerous network management computer systems, each running its own manager software and to have agent processes in the network numbering in the thousands *[column 5, lines 37-41]*. Maddocks teaches that a need exists for a convenient and expedient way to view and analyze machine control sequences and the corresponding execution results *[paragraph 4, lines 1-3]*. By first sending a signal to the agent before execution begins, the system of Bonnell would be able to predetermine which agents are available for execution of commands and which are not. This would allow only the monitoring of agents that are available, which would result in a more organized and expedient method for analyzing machine control sequences over the network. It would have been obvious to one of ordinary skill in the

art at the time the invention was made to combine the teachings of Maddocks with the teachings of Bonnell so that the analysis of machine control sequences could be run on a plurality of remote computers wherein the system determines which remote computers are available before execution.

9-9. Regarding claim 32, Maddocks and Bonnell teach the invention substantially as claimed. Maddocks and Bonnell teach a GUI that displays the operating system commands, the list of remote computers, and the indication that the remote computer is unavailable. See sections 9-7 and 9-8. Maddocks and Bonnell do not expressly teach displaying the information in separate windows. However, Examiner takes Official Notice that the use of separate windows to display information is well known, such as pop-up windows. This allows the user more control over the information displayed within the separate windows, independent of the information within the original window.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to display the operating system commands, list of remote computers, and indication that the remote computer is unavailable in separate windows since Examiner takes Official Notice that the use of separate windows to display information is well known and allows the user more control over the information displayed within the separate windows, independent of the information within the original window.

10. Claims 25, 13, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maddocks et al (Pub. No. US2004/0201627 A1) and Bonnell et al (US Patent No 5,655,081), and further in view of Foster et al (US Patent No 6,684,260).

Claim 25 (Method)

Claim 13 (System)

Claim 29 (Computer Readable Medium)

10-1. Regarding claims 25, 13, and 29, Maddocks and Bonnell teach the invention substantially as claimed. See section 9-1. Maddocks and Bonnell do no expressly teach associating at least one computer operating system command within the plurality of computer operating system commands with one of at least one undo command, displaying, accepting a selection, and executing a selected undo command.

Foster teaches a method for performing consistency checks after commands are given to device drivers *[column 3, lines 8-14]*. Foster teaches the method comprising associating each of at least one computer operating system command within the plurality of computer operating system commands with one of at least one undo command, by teaching, "If neither the "Cancel" *[figure 3, reference character 306]* nor the "OK" *[figure 3, reference character 310]* command button has been activated, the user input event is understood to include one or more modified or selected values for one or more attributes as indicated in the dialog box. At step "320" *[figure 3]* device driver "211" *[figure 1]* obtains from the message obtained at step "304" *[figure 3]* the identifier of the attribute(s) affected by the user input event. The current value(s) for

each identified attribute is then obtained, and at step "322" *[figure 3]* is posted on undo list "226" *[figure 2]*. An identifier of the attribute to which each value is associated is also posted on undo list "226" *[figure 2]*" *[column 11, lines 5-14]*. Thus, the undo list has an undo command associated with the set of computer operating system commands represented by the modified or selected value.

Foster teaches the method comprising displaying the at least one undo command along with an identification of an associated computer operating system command to which each of the at least one undo command is associated, by disclosing a client user interface session including a unique instantiation of user interface process, validation process, undo list, review list, and client's settings *[column 7, lines 49-52]*. *[Figure 2]*, reference character "226" shows the undo list within the client user interface.

Foster teaches the method comprising accepting and executing a selection of a selected undo command, by disclosing in response to user input received at box "356" *[figure 5]*, one or more user-directed modifications will be reversed *[column 14, lines 27-28]*. "The contents of the undo list may include values for attributes which the user has modified, for example, as posted at step "322" *[figure 3]*; or, attributes and values as posted during operation of any selected rule, for example as posted at step "360" *[figure 5]*. At step "358" *[figure 5]* it is preferred to reinstate the attribute values that existed prior to receipt of the user event indicated at step "304" *[figure 3]*. Note that if the user input event at step "304" *[figure 3]* has been validated by a complete operation of step "340" *[figure 3]*, then the scope of the undo operation at step "358" *[figure 5]* corresponds to undoing one user input even cycle *[column 14, lines 32-42]*. Thus, the

user can accept a selection of a selected undo command in response to step "356" *[figure 5]* whereby selecting cancel will execute the undo command.

It would have been obvious to one of ordinary skill in the art at the time the invention was made, to include in the network management computer system for managing commands to be executed on a plurality of server computer systems as taught by Maddocks and Bonnell, the use of an undo command as taught by Foster. This would allow the network management system to undo any unwanted changes made to the remote computer systems that are caused by the executed commands.

Response to Arguments

11. The Examiner acknowledges the Applicants' amendments to claims 1, 6, 7, 9, 13, 15-17, and 22-23, the cancellation of claims 2, 5, 8, 10, 14, 18, 21, and 24, and the addition of new claims 25-32. Regarding independent claims 1, 9, and 17, the Applicants allege that Maddocks (Pub. No. US2004/0201627 A1 to Maddocks et al.) as described in the previous Office action, does not explicitly teach displaying a data file that contains a textual representation corresponding to a computer operating system command and concurrent execution of an operating system command on more than one computer, as has been added to each of these claims. In support of this allegation, the Applicants submit that the claimed invention comprises displaying on a local computer, a data file that contains a plurality of textual representations, each textual representation corresponding to one computer operating system command of the plurality of computer operating system commands and sending the selection of at least

one computer operating system command to a plurality of remote computers communicatively coupled to the local computer via a computer network for concurrent execution of the selection of at least one computer operating system command on at least two of the plurality of remote computers. The Examiner appreciates the Applicants' attempt, via the submitted amendments, to differentiate the claimed invention from that of Maddocks. However, the Examiner respectfully submits that claims 1, 9, and 17, even as amended, fail to present a novel invention, as is evidenced by the fact that Maddocks, in view of Bonnell (US Patent No 5,655,081 to Bonnell et al) is considered to read upon these claims. Maddocks does teach the data file by disclosing, "All of the information needed to enable tests to be set up and executed may be stored in a memory device comprised by a computer" *[paragraph 17, lines 1-3]*. This would "enable multiple users to create, edit, and/or execute machine control sequences at their workstations by accessing previously created machine control sequences" *[paragraph 17, lines 11-14]*. Since all of the information needed to enable tests is stored, users are able to retrieve any part of the information needed to create their own tests. Hence, the commands associated with the devices when the device node is selected *[paragraph 22, lines 1-3]* are stored on a data file and are pre-determined. In this way, the GUI displays a command script file containing a textual representation corresponding to an operator defined plurality of computer operating system commands. Regarding the concurrent execution of an operating system command on more than one computer, Bonnell discloses a system for monitoring and managing a plurality of remote computers using a computer management system located on a local computer, as

discussed in section 9-1 of this Office action. The execution of commands on the remote computers are done independently of the other computers on the network. Thus, as discussed in section 9-1 of this Office action, it would be more efficient to execute commands on multiple remote computers concurrently. Both the inventions of Maddocks and Bonnell are used for testing and monitoring the status of computers. When in combination, given the motivation explained in section 9-1, Maddocks and Bonnell teach the concurrent execution of an operating system command on more than one computer. Consequently, and given the broadest, most reasonable interpretation of their claim language, Maddocks and Bonnell are considered to anticipate claims 1, 9, and 17.

The Examiner acknowledges the Applicants' amendments to claims 6-7, 15-16, and 22-23, and the cancellation of claims 5, 14, and 21. Regarding claims 6-7, 15-16, and 22-23, the Applicant states that the dependent claims recite all the limitations of the independent claims, and thus, are allowable in view of the remarks set forth regarding independently amended claims 1, 9, and 17. However, as discussed above, Maddocks and Bonnell are considered to anticipate claims 1, 9, and 17, and consequently, claims 6-7, 15-16, and 22-23 are anticipated.

Examiner acknowledges the Applicants' amendments to claim 13, however, claim 13 was amended to correct a minor informality and not for patentability or to further limit the claim in view of any prior art. Applicant states that the dependent claim recites all the limitations of the independent claims, and thus, are allowable in view of the remarks set forth regarding independently amended claim 9. However, as discussed

above, Maddocks and Bonnell are considered to anticipate claim 9, and consequently, claim 13, which had been rejected by Maddocks and Foster (US Patent No 6,684,260) in the previous Office action, is anticipated by Maddocks, Bonnell, and Foster as discussed in section 10-1 above.

Examiner acknowledges the addition of new claims 25-32 by the Applicants'. These claims are anticipated by Maddocks, Bonnell, and Foster as discussed in sections 9-7 and 10-1.

The Applicants' arguments have thus been considered, but are not persuasive.

Conclusion

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alvin H. Tan whose telephone number is 571-272-8595. The examiner can normally be reached between 8:30am-4:30pm, Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on 571-272-4048. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AHT
Assistant Examiner
Art Unit 2173

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2173